arXiv:2007.09860v1 [cs.CV] 20 Jul 2020

Learning Gaussian Instance Segmentation in Point Clouds

Shih-Hung Liu[1], Shang-Yi Yu[1], Shao-Chi Wu[1], Hwann-Tzong Chen[1], and Tyng-Luh Liu[2,3]

[1] National Tsing Hua University, Taiwan
[2] Academia Sinica, Taiwan
[3] Taiwan AI Labs

Abstract. This paper presents a novel method for instance segmentation of 3D point clouds. The proposed method is called Gaussian Instance Center Network (GICN), which can approximate the distributions of instance centers scattered in the whole scene as Gaussian center heatmaps. Based on the predicted heatmaps, a small number of center candidates can be easily selected for the subsequent predictions with efficiency, including *i*) predicting the instance size of each center to decide a range for extracting features, *ii*) generating bounding boxes for centers, and *iii*) producing the final instance masks. GICN is a single-stage, anchor-free, and end-to-end architecture that is easy to train and efficient to perform inference. Benefited from the center-dictated mechanism with adaptive instance size selection, our method achieves state-of-the-art performance in the task of 3D instance segmentation on ScanNet and S3DIS datasets. The GICN code is available at https://github.com/LiuShihHung/GICN

Keywords: 3D instance segmentation

1 Introduction

Modeling 3D scenes requires a compilation of vision techniques to solve the corresponding tasks at different levels, such as depth estimation, feature extraction [34,35], planar reconstruction, object detection [32,33,37,43], and 3D semantic/instance segmentation [19,38,41,42]. Among these tasks, 3D instance segmentation is situated at a higher level but is no less challenging. The goal is to segment out each individual object in a 3D scene and assign it a correct class label. Lower-level tasks can be incorporated as components into the pipeline of 3D instance segmentation, and therefore provide different directions for possible improvements.

In this paper, we aim to tackle 3D instance segmentation from the aspects of predicting the probability heatmaps of instance centers and sizes. We propose a center-dictated mechanism to localize each target instance in the point cloud based on the predicted probability heatmaps. Unlike most of the previous 3D instance segmentation methods that rely on box proposals or a predefined set of anchors, our new method can adapt to the context of the scene for predicting a

small number of instance centers directly from the point cloud to produce the final instance bounding boxes and masks.

More specifically, this work addresses the problem of 3D instance segmentation by formulating a task to learn Gaussian instance segmentation. The proposed method, which is called Gaussian Instance Center Network (GICN), is trained to predict a Gaussian heatmap that characterizes the instance centers, as shown in Fig. 1. Such formulation and design are new and particularly beneficial in that we do not need to rely on a predefined set of anchor boxes, nor do we need to generate box proposals that entail further non-maximum suppression. Working on the center heatmap also allows a more intuitive way to visualize and evaluate the intermediate result of training, which is nontrivial for other methods with entwined architectures and implicit mechanisms. See Fig. 2 for example. We can easily compare the predicted heatmap with the ground-truth heatmap and identify the issues for further improvements.

From the predicted heatmap, we can simply select a small set of center candidates to proceed. The high computational cost that hinders point-cloud processing can therefore be greatly reduced. Subsequently, GICN predicts the instance size of each center to determine a proper neighborhood for feature extraction. Based on the size-aware, adaptively extracted features, GICN can better estimate the bounding box and mask for each instance center. As a result, GICN provides a more intuitive 3D pipeline that is easy to train and efficient to perform inference. Our experiments show that the proposed method can achieve state-of-the-art performance on 3D instance segmentation benchmarks. We also conduct ablation study to verify the effectiveness of our design of GICN.

2 Related Work

While the performance of 2D instance segmentation has been significantly advanced by a series of recent work [1,4,6,14,16,20,25,26,27], research on 3D instance segmentation has not yet achieved comparable success as its 2D counterpart. Recent methods for 3D instance segmentation can be mainly characterized into two categories according to their representations: *voxel based* versus *point-cloud based*. Voxel-based methods, such as MTML [19], MASC [24], and PanopticFusion [28], are designed to work on volumetric data, where the 3D space is voxelized into voxel grids for deriving the input representation from scene geometry. On the other hand, point-cloud based methods directly take the point cloud as input and extract features from the 3D points for predicting instance segmentation, *e.g.*, SGPN [38], 3D-BoNet [41], GSPN [42], and others [2,10,31,39]. Furthermore, point-cloud based methods often include PointNet [34] or PointNet++ [35] as the backbone to extract local and global features. In this work, we also use PointNet++ to compute features of 3D points and do not go into developing new methods for 3D feature extraction. The mechanism of feature extraction is orthogonal to the gist of our method, and GICN will also benefit from further improvements in point cloud features.

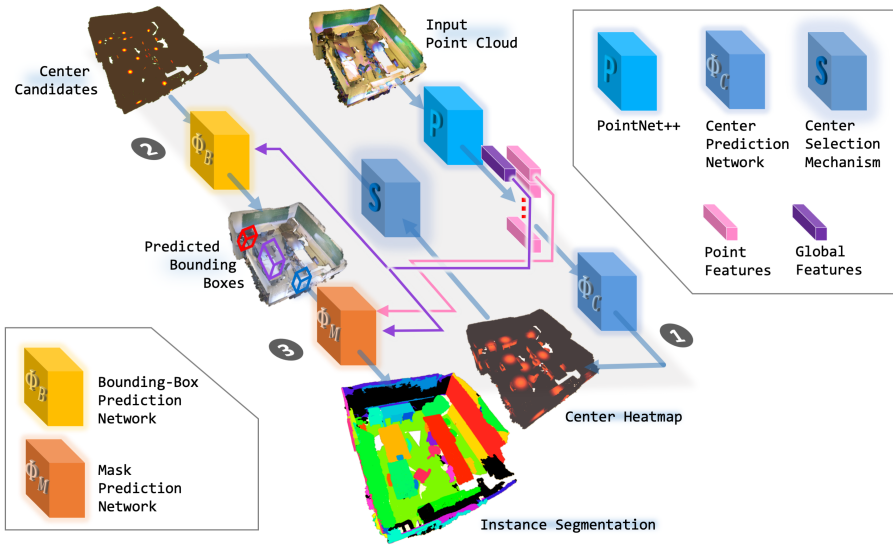

Fig. 1: An overview of GICN. The global and local features are extracted from the input point cloud and then passed through the center prediction network (①) to generate the Gaussian approximation heatmap. We use a center selection mechanism to choose a small number of probable candidates, which will yield the bounding boxes and the instance masks using the bounding-box prediction network (②) and the mask prediction network (③)

Previous ideas that have been shown to be effective for 2D instance segmentation can also be applied to 3D instance segmentation. For example, metric learning is one of the key building blocks in many 2D instance segmentation methods [5,18,21,29,30]. A common metric-learning strategy used in 2D instance segmentation is to define a pairwise loss for learning suitable pixel embeddings, such that, in the embedding space, points belonging to the same instance are drawn closer to each other. For 3D instance segmentation, similar strategy can be applied to the learning of embeddings for 3D points [19,31,39]. For instance, MTML [19] builds upon [5] to learn inter-instance and intra-instance relations, and it adopts a post-processing step for semantic segmentation, using mean-shift clustering [13] to group the 3D points in the embedding space. ASIS [39] and JSIS3D [31] also use mean-shift clustering to obtain instance segmentation clusters from embeddings. Another strategy related to metric learning is to train a network that can directly estimate the instance affinity score for predicting whether two points belong to the same object instance, *e.g.*, MASC [24].

3D instance segmentation is also related to the tasks of 3D semantic segmentation [7,8,11,15,40] and 3D object detection [32,33,37,43]. 3D semantic segmentation is to predict semantic labels for the 3D points, but it does not separate different instances. On the other hand, 3D object detection estimates the 3D bounding box of each individual object, but it is not able to provide

a detailed mask on the 3D points of the target object. Therefore, 3D instance segmentation can be considered as an integrated task of 3D object detection and semantic segmentation, although simply concatenating them together might not yield an effective model to achieve good results.

Yang *et al*. [41] propose an anchor-free approach called 3D-BoNet, which achieves good performance in 3D instance segmentation and shows the advantage of anchor-free prediction. Our method significantly differs from theirs in the design and the methodology. 3D-BoNet predicts a fixed number of 3D bounding boxes and the corresponding instance masks, while our method works on a probability heatmap that can be used to predict an arbitrary but small number of instance centers. Moreover, our method learns to select a suitable instance size for the cluster of points that belong to the same instance center.

The recent method VoteNet [32] for 3D object detection presents a Hough voting mechanism to generate new points that lie close to object centers. The votes are then aggregated into clusters from which box proposals can be derived. Despite the distinction in the tasks to be solved (instance segmentation versus object detection), our method has other fundamental differences from VoteNet. These differences also highlight the advantages and contributions of our work.: *i*) Our method predicts a Gaussian approximation heatmap for centers from the entire point cloud while VoteNet samples a set of seeds to vote to centers. *ii*) Our method can adapt to the distribution of point cloud to decide appropriate cluster sizes for inferring the bounding boxes, while VoteNet sets a fixed aggregation radius for all centers. *iii*) VoteNet generates a fixed number of box proposals and has to perform non-maximum suppression to get the final output bounding boxes. Our method is able to produce the 3D masks immediately from the heatmap and does not need further non-maximum suppression over bounding boxes.

3 Our Method

As illustrated in Fig. 1, the proposed method, Gaussian Instance Center Network (GICN), learns to carry out the task of 3D instance segmentation by first predicting the distribution of instance centers. The strategy is fundamentally different from most of the existing techniques that begin by focusing on predicting bounding-box proposals. Based on a relatively small set of selected center candidates, our method then estimates the corresponding bounding boxes and instance masks. In the following sections, we detail the three key stages to accomplish the proposed Gaussian instance segmentation: *center prediction network* (Sec. 3.1), *bounding-box prediction network* (Sec. 3.2), and *mask prediction network* (Sec. 3.3).

3.1 Center Prediction Network Φ_C

Let $\mathcal{P} = \{p_i = (x_i, y_i, z_i)\}_{i=1}^{N}$ be an input point cloud containing N points. The center prediction network Φ_C is constructed to estimate the probability of each point $p_i \in \mathcal{P}$ being the center of some relevant instance in the 3D scene.

We adopt PointNet++ [35] as the backbone for feature extraction, and obtain the global feature vector and point-wise feature vectors of $\mathcal{P}$. Our center prediction network Φ_C has a similar architecture as PointNet++ with four additional fully-connected layers. Note that each output unit of the last fully-connected layer of Φ_C is converted to probability via sigmoid. Let $Q = \{Q_i\}_{i=1}^{N}$ be the 'heatmap' generated by Φ_C, where $Q_i \in [0, 1]$ is the estimated probability of point p_i being the center of a 3D object instance. For the training of Φ_C, we assume that the points of each object instance are distributed as a 3D multivariate Gaussian, and we derive the continuous relaxations from the discrete instance labels as the heatmap ground truths. Specifically, for each object instance, we pick the point that is closest to the instance's centroid as the Gaussian center, and generate the ground-truth heatmap values by computing the distances from points to the center. We apply a Gaussian function to each distance and normalize the value to $[0, 1]$. Fig. 2 shows two examples of heatmaps predicted by the trained Φ_C on the new input point clouds from the validation set of ScanNet dataset [9]. In comparison with the ground-truth heatmaps, we can see that Φ_C is able to approximate the center distributions very well even for unknown 3D scenes.

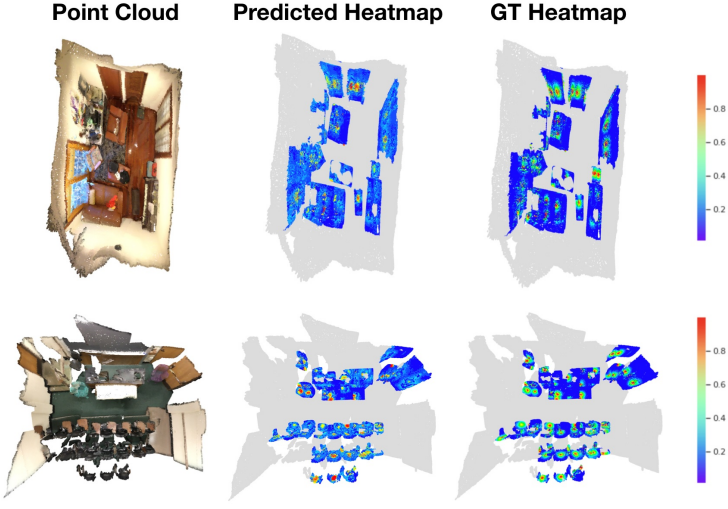

Fig. 2: Visualization of predicted and ground-truth center heatmaps on ScanNet

Center selection mechanism. To choose center candidates from Q, we can simply sort the heatmap values $\{Q_i\}_{i=1}^{N}$ in descending order and pick those points with high probability values as the possible centers. However, since the high-probability points of the same instance tend to form a cluster near the center, naively deciding the center candidates according to a fixed sorting order would result in repeatedly selecting points from the same instance. To overcome this issue, each time a point $p_i \in \mathcal{P}$ is selected as a center candidate, the remaining

Algorithm 1: Center selection & thresholds Q_θ , T_θ

Input: A semantic net Φ_S, representative class radii $\{r_\ell\}$
Output: A set of center candidates $\mathcal{C} = \{C_t\}_{t=1}^{T}$
Data: Point cloud $\mathcal{P} = \{p_i\}_{i=1}^{N}$ and heatmap Q

$\mathcal{C} \leftarrow \emptyset$; $T \leftarrow 0$
$L = \{\ell_i\}_{i=1}^{N} \leftarrow \Phi_S(\mathcal{P})$ // Semantic label
repeat
  $i^* \leftarrow \arg\max_i \{Q_i\}$; $Q^* \leftarrow Q_{i^*}$ // The largest heatmap value
  $T \leftarrow T + 1$; $C_T \leftarrow p_{i^*}$ // A chosen center candidate
  $I^- \leftarrow \{i \,|\, d(p_i, p_{i^*}) \leq r_{\ell^*}\}$ // Filtered by the class radius
  for $i \in I^-$ **do**
    $Q_i \leftarrow 0$ // Excluding redundant points
until $Q^* < Q_\theta$ *or* $T > T_\theta$

sorted list will be updated so that for those high-probability points belonging to the same instance as p_i, their heatmap values have to be reduced to zero. To do so, we train a coupled semantic network (in this work we use *sparse convolution network* [15]) to predict the semantic label of each point p_i and subsequently decide the *representative* class radius r_ℓ for each object class ℓ. Therefore, whenever a point p_i is chosen as a center candidate, the heatmap values of all remaining points within the corresponding radius will be set to zero. In this way the problem of redundant selections can be largely alleviated. The representative radius for each class is defined by the average (class-wise) instance size from the training data.

With $\{Q_i\}_{i=1}^{N}$, the process of center selection will be repeatedly carried out until the currently largest heatmap value of being an instance center is below a pre-specified threshold Q_θ, or the total number T of selected points exceeds T_θ, which is the default upper bound of the number of object instances in a 3D scene. We outline the steps of the center selection mechanism in Algorithm 1, where the two selection thresholds are set as $Q_\theta = 0.4$ and $T_\theta = 64$ for all our experiments. The center selection mechanism will therefore yield $T \leq T_\theta$ center candidates, denoted as $\mathcal{C} = \{C_t\}_{t=1}^{T}$.

We remark that the effect of the proposed center selection mechanism in Algorithm 1 is analogous to performing non-maximum suppression (NMS) in advance. The resulting center candidates would be well separated from each other by the constraint of class radii, leading to less-redundant predicted bounding boxes and instance masks for further processing. Hence, our method does not require post-processing of non-maximum suppression to remove overlapped bounding boxes or masks for instance segmentation.

3.2 Bounding-Box Prediction Network Φ_B

The center prediction network and center selection mechanism provide the set of T predicted instance centers with their 3D coordinates, *i.e.*, $\mathcal{C} = \{C_t\}_{t=1}^{T} \in \mathbb{R}^{T\times 3}$.

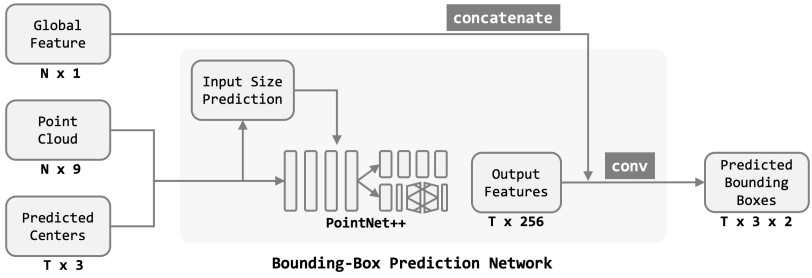

Fig. 3: Bounding-box prediction network Φ_B. The network first predicts the instance size for each of the T selected centers, and then uses a shared PointNet++ network to extract features from the point cloud within the neighborhood of the predicted size. The extracted local features combined with the global features will go through convolutional layers to predict T bounding boxes

In principle, to generate a proper bounding box for each center C_t, we should pay more attention to a properly-sized neighborhood of C_t in the point cloud. To this end, the instances of different classes in the training data are divided into K groups ($K = 6$ in our experiments), and we average the bounding-box sizes of each group to respectively obtain K different typical instance sizes s_k for $k = 1, 2, \cdots, K$, each with different length, width, and height to approximate the predicted instances shapes. Now, to generate the bounding box for each center $C_t \in \mathcal{C}$, we feed the point cloud $\mathcal{P}$ and the center candidates $\mathcal{C}$ to the bounding-box prediction network Φ_B (comprising PointNet++ as the backbone). The network first predicts the probability P_{s_k} that measures how likely the resulting bounding box of C_t could have the instance size s_k. For each center C_t, the network Φ_B processes a context within the most appropriate instance size s_{k^*}, and then uses a shared PointNet++ network to extract local features, which are combined with the global feature for the subsequent convolutional layers to predict the corresponding bounding-box vertices:

$$B_t = \left\{ \left(x_t^{\min}\ y_t^{\min}\ z_t^{\min}\right), \left(x_t^{\max}\ y_t^{\max}\ z_t^{\max}\right) \right\}, \text{ for each selected center } C_t \in \mathcal{C}\,. \tag{1}$$

3.3 Mask Prediction Network Φ_M

Inspired by the effectiveness of the 3D-BoNet [41], we design the mask prediction network Φ_M by simultaneously considering point-wise and global features to predict the instance masks based on the resulting bounding boxes in the previous stage. However, a crucial difference between 3D-BoNet and our method is that we would consider T center candidates to predict T bounding boxes, and use these bounding boxes to predict T instance masks. The number T is not fixed and can adapt to each scene, depending on how many center candidates are uncovered by the center prediction network, while 3D-BoNet always handles a fixed predefined number of bounding boxes.

3.4 Loss Functions

The proposed network is trained in an end-to-end manner and optimized by a joint loss $\mathcal{L}_{\text{total}}$ consisting of several loss terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{center}} + \mathcal{L}_{\text{bound}} + \mathcal{L}_{\text{IoU}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{size}} \,. \tag{2}$$

Specifically, we use the focal loss [23] for both $\mathcal{L}_{\text{center}}$ and $\mathcal{L}_{\text{mask}}$ to enhance the center prediction network Φ_C and the mask prediction network Φ_M by focusing more on a sparse set of hard examples during training.

The center loss term $\mathcal{L}_{\text{center}}$ for center prediction network Φ_C is defined as

$$\mathcal{L}_{\text{center}} = \sum_{i=1}^{N} -\alpha(1 - Q_{i,f})^{\gamma} \log(Q_{i,f}) \,, \tag{3}$$

where α and γ are the focal loss parameters, and f symbolizes the focal setting. For each point p_i, we set $Q_{i,f} = Q_i$ if $\widehat{G}_{t(i)}(p_i \,|\, \widehat{C}_{t(i)}) > \sigma_G$, where $\widehat{G}_{t(i)}$ is the Gaussian on the ground-truth instance center $\widehat{C}_{t(i)}$ that is closest to p_i. We use $\sigma_G = 0.4$ in our experiments. For the other case with $\widehat{G}_{t(i)}(p_i \,|\, \widehat{C}_{t(i)}) \leq \sigma_G$, we set $Q_{i,f} = 1 - Q_i$, which means p_i is not considered to be associated with the closest ground-truth instance.

To learn feasible instance sizes and bounding boxes from the predicted centers in the bounding-box prediction network Φ_B, we define the sizes loss $\mathcal{L}_{\text{sizes}}$ as the cross entropy loss and the bounding-box bound loss $\mathcal{L}_{\text{bound}}$ as l_1 loss. Specifically, we can express the two losses as

$$\mathcal{L}_{\text{size}} = \sum_{t=1}^{T} -\log(S_t) \,, \tag{4}$$

$$\mathcal{L}_{\text{bound}} = \frac{1}{T} \sum_{t=1}^{T} l_1^{\text{smooth}}(B_t - \widehat{B}_t) \,, \tag{5}$$

where T is total number of predicted centers and thus also the number of predicted bounding boxes; B_t contains the vertices of the predicted bounding box for center C_t; $\widehat{B}_t$ contains the vertices of the corresponding ground-truth bounding box. S_t is the predicted sizes probability, which assumes its value from one of P_{s_k} for $k = 1, 2, \cdots, K$, depending on the size value of the corresponding ground truth. We use the smooth l_1 loss rather than the l_2 loss to ensure training stability and convergence.

Notice that, in comparison with the multi-criteria loss employed by 3D-BoNet [41] for box prediction, which needs a box association layer to decide the mapping between predicted and ground-truth bounding boxes, our bounding box loss $\mathcal{L}_{\text{bound}}$ in (5) is much simpler and more intuitive. Since the proposed GICN uses center candidates to predict bounding boxes, each bounding box exactly corresponds to a predicted center. Thus, the predicted bounding box can be

conveniently associated with the ground-truth bounding box for the computation of the smooth l_1 loss.

Further, we use GIoU [36] instead of vanilla IoU to compute the IoU loss, which is defined as

$$\mathcal{L}_{\text{IoU}} = \frac{1}{T}\sum_{t=1}^{T}\left(1 - \text{GIoU}(B_t, \widehat{B}_t)\right) . \tag{6}$$

Finally, as mentioned early, we use the focal loss for our mask loss term $\mathcal{L}_{\text{mask}}$ to compute the loss between the ground-truth and the predicted mask probability for each instance mask.

4 Experiments

4.1 Datasets

In the experiments we evaluate the performance of the proposed method on two benchmark datasets. Both datasets provide 3D data in form of *colored* point clouds that consist of 3D coordinates and RGB color information for the 3D scenes. The two datasets are detailed as follows.

- **Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset** [3] collects six large-scale indoor-area scans from 271 rooms in three different buildings. We take the standard k-fold cross-validation scheme to evaluate the validation performance on S3DIS dataset.
- **ScanNet dataset** [9] is an RGB-D large-scale dataset containing 1,513 scans annotated with instance-level semantic segmentation labels. We randomly take 1,201 scenes for training and 312 scenes for validation, and finally test our method on ScanNet online benchmark for final evaluation.

In the following sections we report our validation performance on S3DIS dataset and the test performance on ScanNet dataset. We also perform the ablation study on Area-5 of S3DIS dataset to investigate the effectiveness of each component in the proposed pipeline.

4.2 Implementation Details

We implement the proposed GICN in PyTorch and use two Nvidia GTX1080Ti GPUs for training. The learning rate of the model is 0.002 and then decays by half the value every 20 epochs. We use Adam optimizer to train the network. Our network usually converges at the 50th epoch, which takes about one day and three days, respectively, for training with S3DIS and ScanNet datasets.

Similar to SGPN [38] and 3D-BoNet [41], during training we divide the whole scene into cubes of 1m^3 volume with a sliding window of stride 0.5m. At the test time, we perform the inference on all cubes in the whole scene, and use the block-merging algorithm as SGPN to merge each cube's result to get the final output

Table 1: Comparisons on S3DIS instance segmentation (6-fold cross validation)

	mPrec (%)	mRec (%)
ASIS [39]	63.6	47.5
3D-BoNet [41]	65.6	47.6
3D-BEVIS [10]	65.6	n/a
GICN (ours)	**68.5**	**50.8**

of instance segmentation for the 3D scene. Regarding the hyperparameter T_θ for the maximum number of instance centers, we set it as 64 for both training and testing. Note that, in practice, after performing the center selection mechanism we usually have only 1 to 5 centers left in a cube during testing.

4.3 Evaluation on S3DIS Dataset

We test our method on S3DIS dataset, in which each scene is partitioned into $1m^3$ cubes, and the number of 3D points of each cube is uniformly sampled to produce 4,096 points for training and testing. Each point is then represented by a 9D vector (RGB, normalized XYZ in block, and normalized XYZ in room) with a label from one of the 13 classes. We use 6-fold cross validation to evaluate the performance, and the scores and qualitative results are shown in Table 1 and Fig. 4.

We compare the proposed GICN with 3D-BoNet (the state-of-the-art method on S3DIS dataset), as well as 3D-BEVIS [10] and ASIS [39]. The metrics we use for the evaluation are mean precision (mPrec) and mean recall (mRec) with IoU threshold 0.5. We use the block-merging algorithm to merge the instances from different cubes like SGPN [38]. Our proposed method outperforms the state-of-the-art methods by at least 2.9% increase in mAP, owing to the formulation and the learning of the Gaussian heatmap that approximates the distribution of instance centers and sizes for generating instance masks.

4.4 Evaluation on ScanNet Dataset

We further evaluate our method on ScanNet v2 3D semantic instance segmentation dataset. Each scene is also divided into $1m^3$ cubes with uniformly sampled 4,096 points for training. Our model is applied to all points during testing, and we use the block-merging algorithm [38] to construct the complete segmentation of the entire 3D scene.

The evaluation is performed on 18 object classes and the average precision with an IoU threshold 0.5 (AP@50%) is used as the evaluation metric. For comparison, we show our quantitative results in Table 2 based on the ScanNet v2 benchmark, and the qualitative results are shown in Fig. 5. The proposed GICN achieves the state-of-the-art performance in comparison with the existing

Table 2: ScanNet v2 instance segmentation online benchmark. The table shows AP@50% score of each semantic class. Our method achieves the best mean AP@50% performance among all existing methods published in the literature

Method	mean	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	desk	door	other	picture	refrigerator	shower curtain	sink	sofa	table	toilet	window
SGPN [38]	14.3	20.8	39.0	16.9	6.5	27.5	2.9	6.9	0.0	8.7	4.3	1.4	2.7	0.0	11.2	35.1	16.8	43.8	13.8
3D-BEVIS [10]	24.8	66.7	56.6	7.6	3.5	39.4	2.7	3.5	9.8	9.9	3.0	2.5	9.8	37.5	12.6	60.4	18.1	85.4	17.1
DPC-instance [12]	35.5	50.0	51.7	46.7	22.8	42.2	13.3	40.5	11.1	20.5	24.1	7.5	23.3	30.6	44.5	43.9	45.7	97.4	23.9
3D-SIS [17]	38.2	**100**	43.2	24.5	19.0	57.7	1.3	26.3	3.3	32.0	24.0	7.5	42.2	85.7	11.7	69.9	27.1	88.3	23.5
MASC [24]	44.7	52.8	55.5	38.1	38.2	63.3	0.2	50.9	26.0	36.1	43.2	32.7	45.1	57.1	36.7	63.9	38.6	98.0	27.6
ResNet-backbone [22]	45.9	**100**	73.7	15.9	25.9	58.7	13.8	47.5	21.7	41.6	40.8	12.8	31.5	71.4	41.1	53.6	59.0	87.3	30.4
PanopticFusion [28]	47.8	66.7	71.2	59.5	25.9	55.0	0.0	61.3	17.5	25.0	**43.4**	**43.7**	41.1	85.7	48.5	59.1	26.7	94.4	35.9
3D-BoNet [41]	48.8	**100**	67.2	59.0	30.1	48.4	9.8	62.0	30.6	34.1	25.9	12.5	43.4	79.6	40.2	49.9	51.3	90.9	43.9
MTML [19]	54.9	**100**	80.7	58.8	32.7	64.7	0.4	**81.5**	18.0	41.8	36.4	18.2	44.5	**100**	44.2	68.8	57.1	**100**	39.6
GICN (ours)	**63.8**	**100**	**89.5**	**80.0**	**48.0**	**67.6**	**14.4**	73.7	**35.4**	**44.7**	40.0	36.5	**70.0**	**100**	**56.9**	**83.6**	**59.9**	**100**	**47.3**

instance segmentation methods that have already been published in the literature at the time of ECCV 2020 submission. It can be seen that our method performs less well on classes of instances that resemble a vertical surface, *e.g.*, curtain and picture, in comparison with other voxel-based methods like MTML [19] and PanopticFusion [28]. Such classes are harder to find centers, while other classes like toilet and bathtub have more compact structures so that their centers are easier to be identified.

4.5 Ablation Study

The ablation study is aimed to investigate the effectiveness of each key component of GICN. We expect to provide more insights into how the components may affect the performance of 3D instance segmentation. We conduct the ablation study on the Area-5 data of S3DIS dataset, which is the hardest area among all areas in S3DIS. Table 3 summarizes the quantitative results of our ablation study.

Table 3: Ablation study on Area-5 of S3DIS dataset (†: random selection. $*$: top T_θ)

	mPrec	mRec
Ours (GICN)	**61.5**	**43.2**
w/o Instance Size Prediction	57.8	41.3
w/o Focal Loss	47.4	35.3
w/o Center Prediction / Selection	$51.2^{\dagger}$ / 53.1^{*}	$32.2^{\dagger}$ / 33.7^{*}
w/o Semantic Radius Prior	59.4	42.1

1. **Without instance size prediction**: GICN predicts the probability of the instance size for modeling each instance center and then checks which size

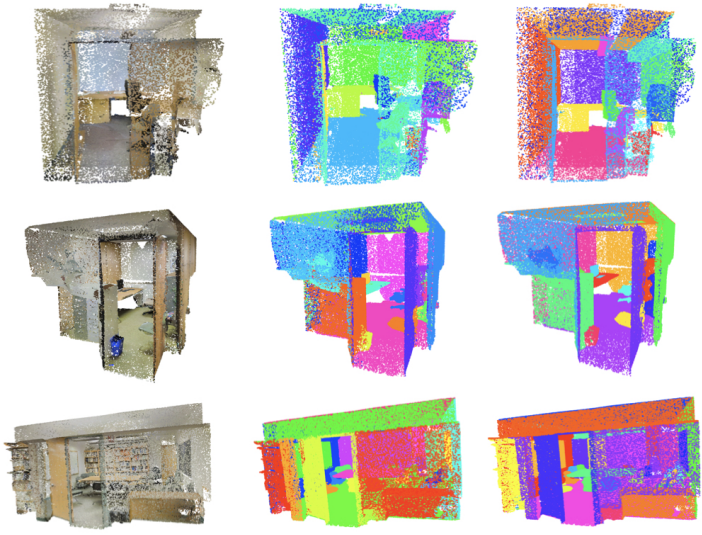

Fig. 4: Results of S3DIS dataset. The first column shows the input point clouds. The second column depicts the predicted masks. The third column shows the ground-truth masks. Note that the color code assigned to each instance does not have to match the ground truth. Only the structure of the mask matters

group the center belongs to for subsequent bounding box prediction. For comparison, we retrain the network to directly predict the bounding box without predicting the instance size, and just extract features from points inside a fixed range. The result shows that instance size prediction improves the performance by 3.7% on mAP.

2. **Without focal loss**: To make GICN focus on difficult cases, we use the focal loss [23] to help the network solve the imbalance problem of predicting various instances. The gap between using the focal loss and using a vanilla cross entropy loss is large. We observe an improvement of 14% on mAP. The

Table 4: The timing results on the ScanNet v2 validation split (312 scenes)

Method	SGPN	ASIS	GSPN
Time (sec)	network (GPU): 650 group merging (CPU): 46,562 block merging (CPU): 2,221	network (GPU): 650 mean shift (CPU): 53,886 block merging (CPU): 2,221	network (GPU): 500 point sampling (GPU): 2,995 neighbor search (CPU): 468
Total Time (sec)	49,433	56,757	3,963

Method	3D-SIS	3D-BoNet	**GICN (ours)**
Time (sec)	voxelization, projection, network, etc. (GPU+CPU): 38,841	network (GPU): 650 SCN (GPU parallel): 208 block merging (CPU): 2,221	network (GPU): 467 SCN (GPU parallel): 208 block merging (CPU): 2,221
Total Time (sec)	38,841	2,871	**2,688**

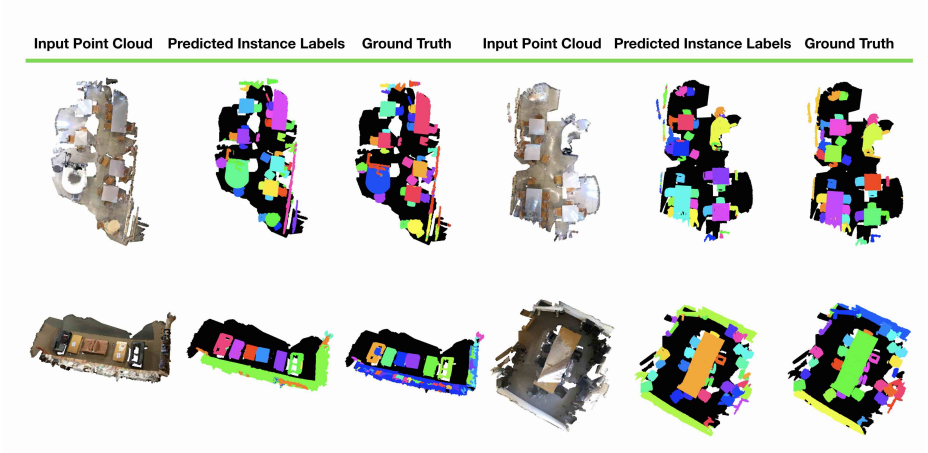

Fig. 5: Qualitative results of the validation split of ScanNet v2 dataset. Different colors indicate different instances. Moe results are in Appendix B

result shows that without the focal-loss strategy our network tends to learn merely the simpler instances.

3. **Without center prediction or selection**: Our center prediction network yields the center heatmap for deciding the center candidates. To validate its effect, we replace the heatmap-guided center selection by randomly choosing T_θ centers as the candidates or by selecting the top T_θ centers based on the heatmap values. (We set $T_\theta = 64$.) The drop of 10.3% and 8.4% on mAP shows that the predicted heatmaps and the selection mechanism provide useful center information for further prediction of bounding box and mask.
4. **Without semantic radius prior**: In the center selection mechanism, we use semantic class radii to choose the center. If we assume that instances of all classes have uniform size, the mAP drops 2.1% and the mRec drops 1.1% because we may redundantly select duplicate centers in one instance and miss instances.

4.6 Discussions

Computational cost. Table 4 summarizes the timing results of different 3D instance segmentation approaches. Experiments are done using a single Titan X GPU. The results show that our method is more efficient than other methods because we do not need additional post-processing steps like Mean Shift or NMS. Moreover, benefited from the center select mechanism, we only need to handle a small number of predicted instances and therefore is faster than 3D-BoNet [41], which predicts a larger, fixed number of bounding boxes.

Dealing with hollow objects. If the shape of an instance is hollow (*e.g.* bathtub), there would be no point cloud near the central region and most of its

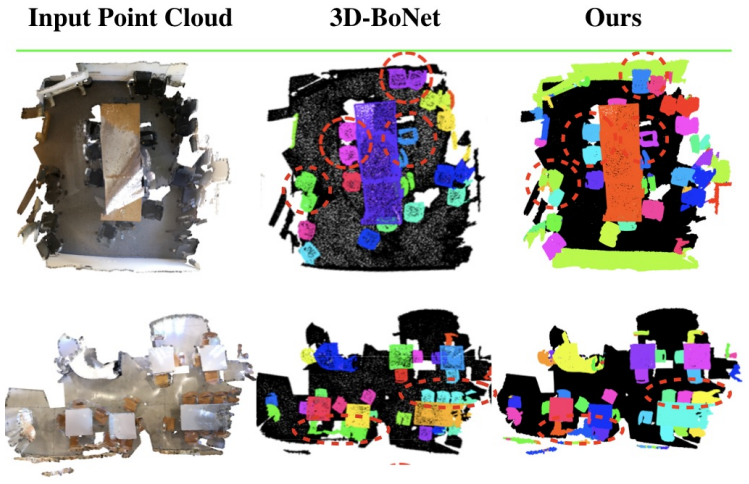

Fig. 6: Comparison with 3D-BoNet from the validation split of ScanNet v2 dataset. The red circles show some examples that 3D-BoNet fails to segment but the proposed GICN successfully produces the instance masks

points might be far from the instance center location. In that case, we will choose the point that is nearest to the center location, and the instance size prediction mechanism described in Sec. 3.2 could estimate an appropriate size that covers most of the point cloud even if the points are not close to the instance center.

The case of two center candidates being close to each other. The distance constraint of the center selection mechanism (Step 6 of Algorithm 1) is imposed to avoid selecting more than one center candidate for the same instance. It rarely happens that the constraint eliminates all the points of a nearby instance since the semantic radius prior is derived from the training data and is therefore quite reliable in principle. For comparison, Fig. 6 illustrates some example results of GICN and 3D-BoNet [41] on the validation split of ScanNet v2 dataset. Our method can separate the instances well even if they are close to each other while 3D-BoNet [41] fails to generate the correct masks.

5 Conclusion

We have presented a novel center-dictated size-aware 3D instance segmentation method on point clouds. The proposed method, Gaussian Instance Center Network (GICN), aims for learning the Gaussian heatmap that approximates the spatial distribution of instances. By leveraging the center prediction mechanism, GICN can extract the precise instance masks according to the information encoded from the localized centers. We demonstrate the ability of GICN by evaluating the validation and test performance on S3DIS and ScanNet datasets. GICN achieves state-of-the-art results on both benchmarks. Future work may include improving

the accuracy of finding centers for those difficult semantic classes, as well as using metric learning like MTML [19] to learn feature embeddings for further enhancement on visual semantic reasoning.

Appendix A: Generating Ground-Truth Center Heatmaps

Our method needs ground-truth center heatmaps to train the center prediction network. In Fig. 7 we show the point cloud of a chair as an example to explain how we generate the ground-truth center heatmaps for training. To compute the heatmap values for an instance in a scene, we first find the point closest to the instance center, and then we apply Gaussian function to all points of the instance with respect to the chosen centroid point and get the final heatmap for the instance.

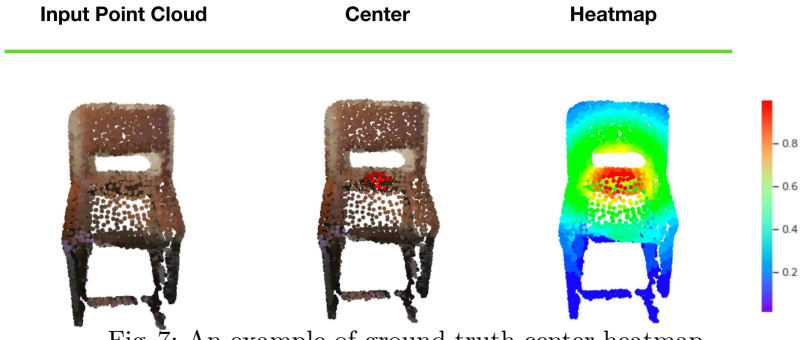

Fig. 7: An example of ground-truth center heatmap

Appendix B: More Qualitative Results on ScanNet and S3DIS Datasets

We show more heatmap predictions and qualitative results on the validation split of ScanNet v2 dataset in Fig. 8 and Fig. 9. Additional results on S3DIS dataset are shown in Fig. 10. From Fig. 8 it can be observed that the center heatmaps are quite capable of capturing instance information, and our method is able to predict accurate center heatmaps in comparison with the ground truth. The peaks in a heatmap imply the center positions, and using the center selection mechanism mentioned in the main paper can easily single out proper centers for further prediction. In the fourth row of Fig. 8 we can see that although some instances are close to each other, the predicted heatmaps can still well represent the center probability.

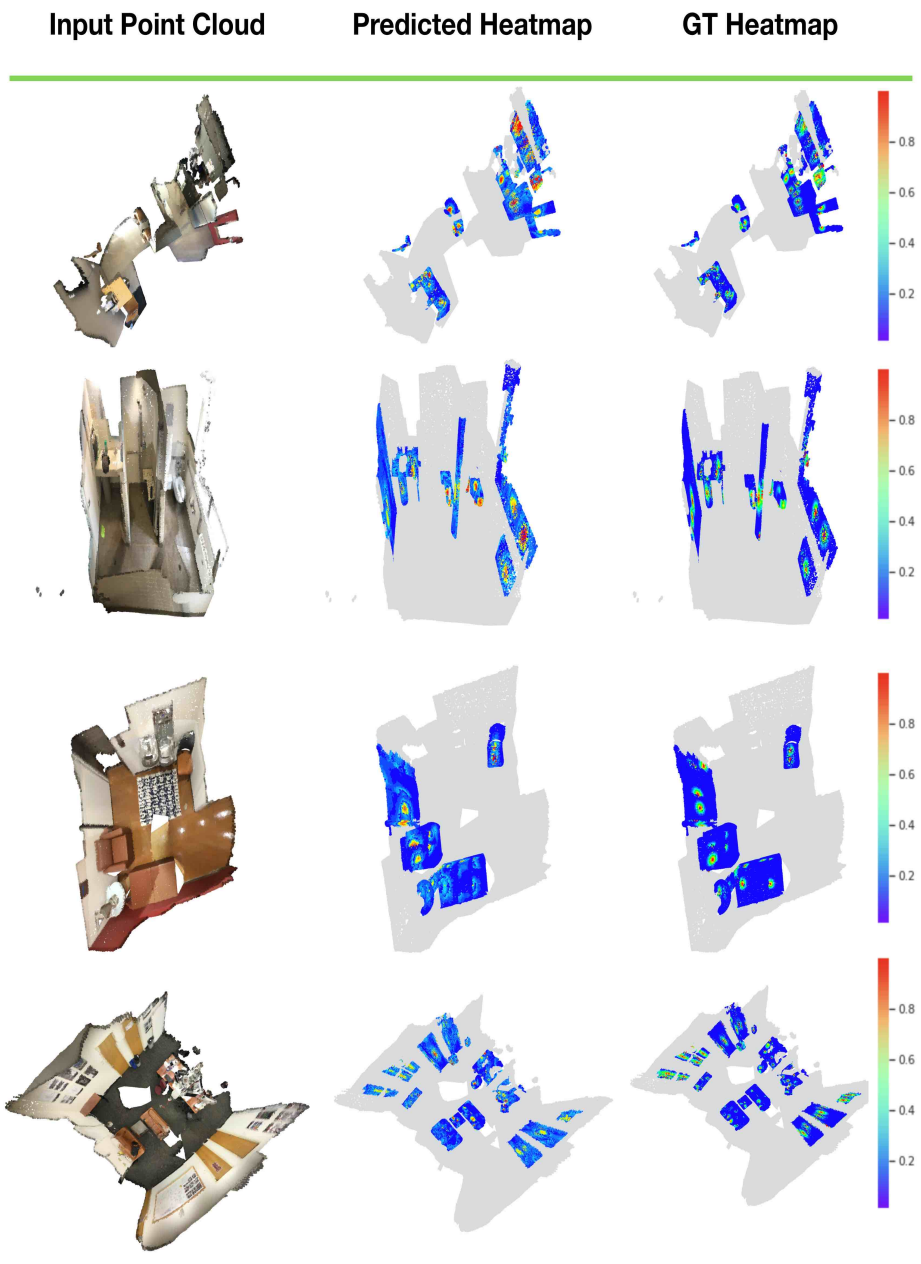

Fig. 8: Center heatmap prediction results from the validation split of ScanNet v2 dataset. The background is shown in gray color

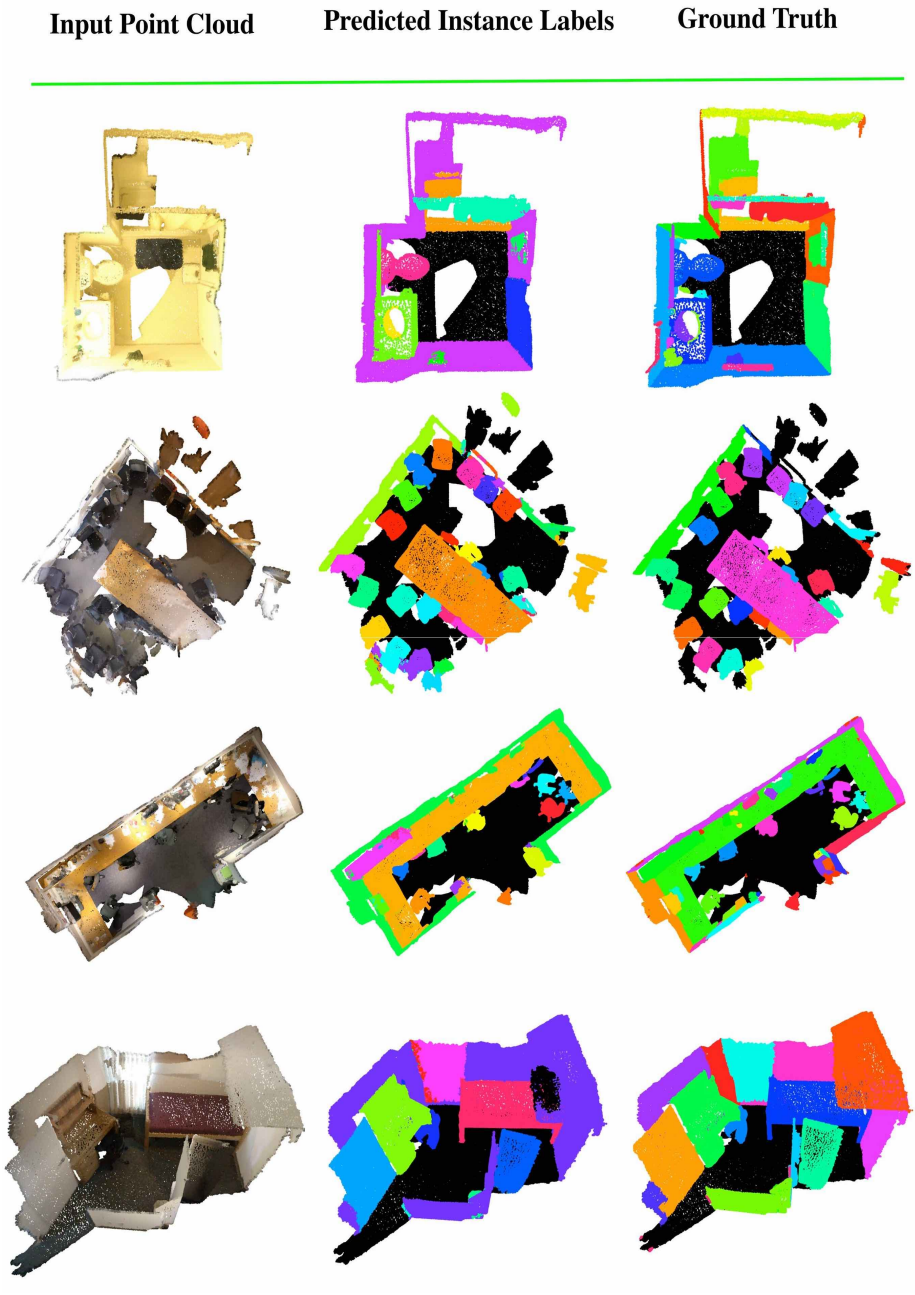

Fig. 9: More qualitative results from the validation split of ScanNet v2 dataset. Different colors indicate different instances. We illustrate the background semantic in black for better visualization

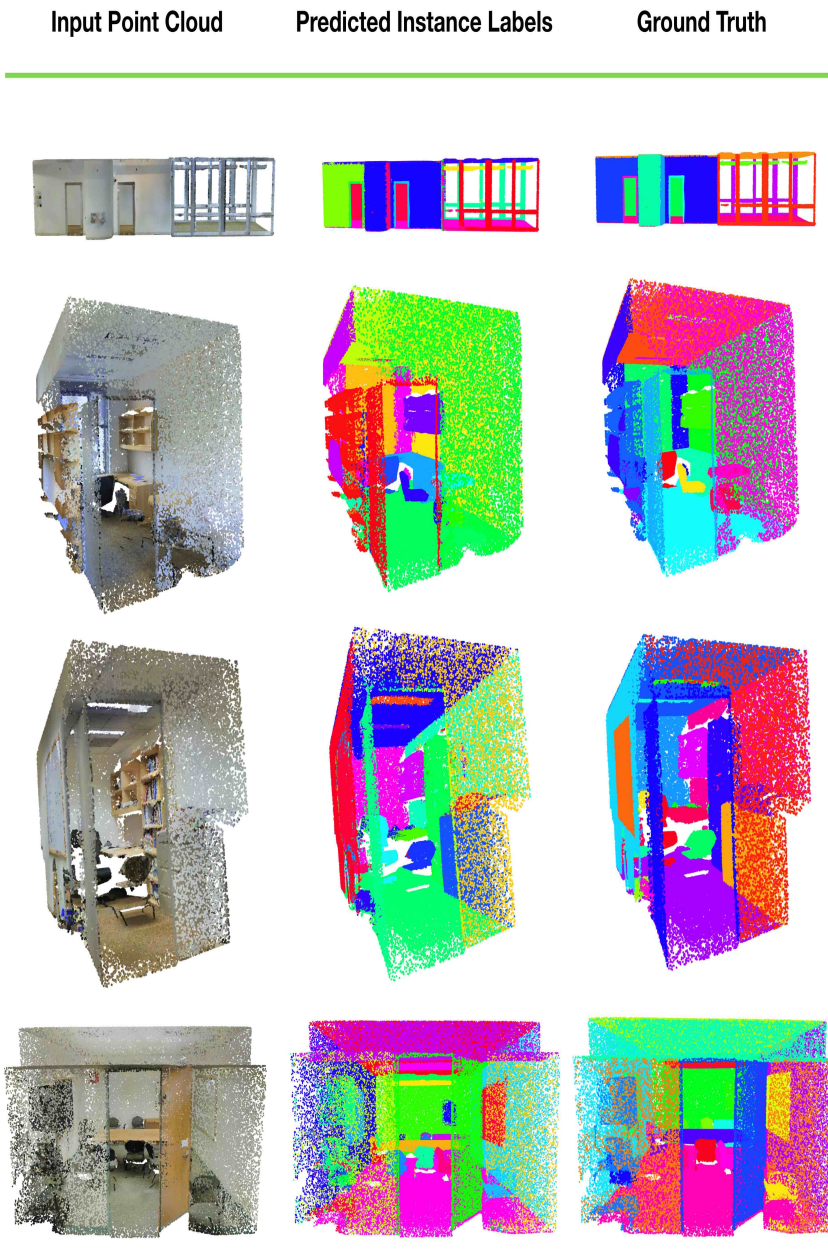

Fig. 10: More qualitative results from the validation split of S3DIS dataset. Different colors indicate different instances

Appendix C: Network Architecture Details

The proposed model, Gaussian Instance Center Network (GICN), consists of three sub-networks: center prediction network Φ_C, bounding-box prediction network Φ_B, and mask prediction network Φ_M. We describe the network architecture of GICN in this section and summarize the details is Table 5.

Table 5: Detailed network architecture of GICN

Sub-network	Architecture
Center prediction network	SA(1024, 0.1, [32, 32, 64])
	SA(256, 0.2, [64, 64, 128])
	SA(64, 0.4, [128, 128, 256])
	SA(None, None, [256, 256, 512])
	FP([256, 256])
	FP([256, 256])
	FP([256, 128])
	FP([128, 128, 128])
Bounding-box prediction network	MLP([64, 128, 256]) (Before concat)
	MLP([512, 128, 6]) (After concat)
Mask prediction network	Conv(64, [1, 134], [1,1])
	Conv(32, [1, 1], [1, 1])
	Conv(1, [1, 1], [1, 1])

For the center prediction network, we use PointNet++ as our backbone. Following the same notation in PointNet++, we have SA($K, r, [l_1, \ldots, l_d]$) as a set abstraction (SA) module that contains d 1×1 convolution layers for K neighbourhood regions in radius r, where l_i ($i = 1, \ldots, d$) is the number of output channels for the ith layer. FP($[l_1, \ldots, l_d]$) is a feature propagation (FP) module consisting of d 1×1 layers, where the ith layer has l_i output channels.

The bounding-box prediction network takes the point cloud and the predicted centers as input, and use PointNet++ to encode the context with respect to the predicted sizes. The parameters used in the bounding-box prediction network are shown in Table 5, where MLP($[l_1, \ldots, l_d]$) comprises multi-layer perceptron with l_i output channels for the ith layer ($i = 1, ..., d$). The output features will be concatenated with global features and then fed into several MLPs to predict the bounding box coordinates.

In the mask prediction network, we use the convolution layer to reduce dimensions of point features and global features, and then concatenate them together. The concatenated features go through several convolution layers with the predicted bounding box information to localize the instances. Finally, these features will pass through three convolution layers listed in Table 5 to get $N \times 1$ mask for each predicted instance. Conv($C, [h, w], [s_1, s_2]$) is a convolution layer, where $[h, w]$ is the kernel size and $[s_1, s_2]$ denotes the stride. Note that the kernel size $[1, 134]$ represents the 128 channels of concatenated features plus the six-dimensional box information.

References

1. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 2209–2218 (2019)
2. Arase, K., Mukuta, Y., Harada, T.: Rethinking task and metrics of instance segmentation on 3d point clouds. CoRR **abs/1909.12655** (2019)
3. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1534–1543 (2016)
4. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 2858–2866 (2017)
5. Brabandere, B.D., Neven, D., Gool, L.V.: Semantic instance segmentation with a discriminative loss function. CoRR **abs/1708.02551** (2017)
6. Chen, L., Hermans, A., Papandreou, G., Schroff, F., Wang, P., Adam, H.: MaskLab: instance segmentation by refining object detection with semantic and direction features. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 4013–4022 (2018)
7. Chiang, H., Lin, Y., Liu, Y., Hsu, W.H.: A unified point-based framework for 3d segmentation. In: 2019 International Conference on 3D Vision, 3DV 2019, Québec City, QC, Canada, September 16-19, 2019. pp. 155–163 (2019)
8. Choy, C.B., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 3075–3084 (2019)
9. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., Savarese, S.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2432–2443 (2017)
10. Elich, C., Engelmann, F., Kontogianni, T., Leibe, B.: 3d bird's-eye-view instance segmentation. In: Pattern Recognition - 41st DAGM German Conference, DAGM GCPR 2019, Dortmund, Germany, September 10-13, 2019, Proceedings. pp. 48–61 (2019)
11. Engelmann, F., Kontogianni, T., Leibe, B.: Dilated point convolutions: On the receptive field of point convolutions. CoRR **abs/1907.12046** (2019)
12. Engelmann, F., Kontogianni, T., Leibe, B.: Dilated point convolutions: On the receptive field of point convolutions (2019)
13. Fukunaga, K., Hostetler, L.D.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Information Theory **21**(1), 32–40 (1975)
14. Gao, N., Shan, Y., Wang, Y., Zhao, X., Yu, Y., Yang, M., Huang, K.: SSAP: single-shot instance segmentation with affinity pyramid. In: IEEE International Conference on Computer Vision, ICCV 2019 (2019)
15. Graham, B., Engelcke, M., van der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 9224–9232 (2018)

16. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2980–2988 (2017)
17. Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3d semantic instance segmentation of RGB-D scans. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 4421–4430 (2019)
18. Kong, S., Fowlkes, C.C.: Recurrent pixel embedding for instance grouping. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 9018–9028 (2018)
19. Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3d instance segmentation via multi-task metric learning. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
20. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 4438–4446 (2017)
21. Liang, X., Lin, L., Wei, Y., Shen, X., Yang, J., Yan, S.: Proposal-free network for instance-level object segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **40**(12), 2978–2991 (2018)
22. Liang, Z., Yang, M., Wang, C.: 3d graph embedding learning with a structure-aware loss function for point cloud semantic instance segmentation (2019)
23. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2999–3007 (2017)
24. Liu, C., Furukawa, Y.: MASC: multi-scale affinity with sparse convolution for 3d instance segmentation. CoRR **abs/1902.04478** (2019)
25. Liu, S., Jia, J., Fidler, S., Urtasun, R.: SGN: sequential grouping networks for instance segmentation. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 3516–3524 (2017)
26. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 8759–8768 (2018)
27. Liu, Y., Yang, S., Li, B., Zhou, W., Xu, J., Li, H., Lu, Y.: Affinity derivation and graph merge for instance segmentation. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III. pp. 708–724 (2018)
28. Narita, G., Seno, T., Ishikawa, T., Kaji, Y.: PanopticFusion: online volumetric semantic mapping at the level of stuff and things. CoRR **abs/1903.01177** (2019)
29. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 8837–8845 (2019)
30. Novotný, D., Albanie, S., Larlus, D., Vedaldi, A.: Semi-convolutional operators for instance segmentation. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I. pp. 89–105 (2018)
31. Pham, Q., Nguyen, D.T., Hua, B., Roig, G., Yeung, S.: JSIS3D: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 8827–8836 (2019)

32. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
33. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from RGB-D data. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 918–927 (2018)
34. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 77–85 (2017)
35. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. pp. 5099–5108 (2017)
36. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 658–666 (2019)
37. Sindagi, V.A., Zhou, Y., Tuzel, O.: Mvx-net: Multimodal voxelnet for 3d object detection. In: International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019. pp. 7276–7282 (2019)
38. Wang, W., Yu, R., Huang, Q., Neumann, U.: SGPN: similarity group proposal network for 3d point cloud instance segmentation. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 2569–2578 (2018)
39. Wang, X., Liu, S., Shen, X., Shen, C., Jia, J.: Associatively segmenting instances and semantics in point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 4096–4105 (2019)
40. Wu, W., Qi, Z., Li, F.: Pointconv: Deep convolutional networks on 3d point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 9621–9630 (2019)
41. Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N.: Learning object bounding boxes for 3d instance segmentation on point clouds. CoRR **abs/1906.01140** (2019)
42. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: GSPN: generative shape proposal network for 3d instance segmentation in point cloud. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 3947–3956 (2019)
43. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 4490–4499 (2018)